Use of Word Level Side Information to Improve Speech Recognition

Dimitra Vergyri

Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD 21218-2686
dverg@clsp.jhu.edu

ABSTRACT

Confidence measures for the output of a speech recognizer have been, for some years now, a topic of interest in the speech community. Initially the main goal was to use them as diagnostic tools to understand recognizer behavior by identifying regions and sources of error. But they have also proven useful in other tasks, such as supervised or unsupervised acoustic model adaptation, confidence conditioned language modeling, semantic analysis, and improving word accuracy through rescoring techniques [3]. In all these cases various features were used to compute a confidence measure for each hypothesized word or sentence [2, 5].

In this paper we use similar features that can be measured for each hypothesized word-token and have been commonly used in the literature to assign confidence. However we use them to improve recognition performance. We treat them as knowledge sources, and combine them linearly, along with the acoustic and language model log-probabilities, to get a new log-likelihood score for each hypothesis. The weights of the log linear sentence model are optimized using minimum word error training in the Discriminative Model Combination (DMC) framework [1]. The test hypotheses are rescored using the new likelihood score. Thus we investigate the ability of the features to predict, not the correctness of a hypothesis (confidence), but the correct hypothesis itself.

Statistically significant improvements in WER are reported on the SWBD corpus.

1. METHOD

1.1. Word-level Feature Extraction

In this work we assume that we have available a word lattice for each utterance, which contains the set of alternative hypotheses $\mathcal{H}$ for the given acoustic signal. The word lattices are obtained using the baseline recognizer.

The features we consider are associated with a hypothesized word token h_i with certain start and end times (s_i, e_i) in the acoustic signal. The triplet $w_i = (h_i, s_i, e_i)$ represents a link in a word lattice produced by the recognizer.

The lattice itself (all the paths in it and the model scores) constitutes an information source from which we can collect features (measurements) for each w_i. We also use other information sources, such as a phone recognizer output for the interval (s_i, e_i), or the dictionary pronunciations of hypothesized word h_i, to obtain features which are not present in the lattice, but can be associated with each lattice link w_i.

We wish to collect features that contain information about how probable it is for w_i to be correct, i.e for the word h_i to appear in the true transcription during the time segment (s_i, e_i). The acoustic and language model scores are obvious features to use for that purpose, since they contain the information the baseline system uses. Measurements that have been used in the literature to predict the correctness of the link (confidence) can also be employed in our method. We can use either local features capturing local information from the segment (s_i, e_i) or global features which incorporate the information from the totality of the hypotheses.

1.2. Model Definition

For every instance of a hypothesized word w_i, realized as link in a word lattice or an N-best hypotheses list, we extract a vector of features $(y_1, ... y_M)$ that describes the information available, denoted as $\mathcal{I}$. These features are used as knowledge sources and their values are combined in a log linear model that computes the probability of w_i given the information available:

$$\log p_\Lambda(w_i|\mathcal{I}) = \sum_{j=1}^{M} \lambda_j y_j - z(\Lambda) \qquad (1)$$

where $z(\Lambda)$ is a normalization factor that guarantees p_Λ is a probability distribution on the domain of w, and $\Lambda = \{\lambda_1, ..., \lambda_M\}$ is the set of weights for the features combined.

For a hypothesis $W = (w_1, \dots, w_R)$ we can define the measurements $Y_j = \sum_{i=1}^{R} y_{ij}$. Assuming that the word-links are conditionally independent given $\mathcal{I}$, then the log probability of W becomes a linear combination of the features Y_j [6]:

$$\log P_\Lambda(W|\mathcal{I}) = \sum_{j=0}^{M} \lambda_j Y_j - C(\Lambda) \qquad (2)$$

where we have added the feature $Y_0 = R$, the number of words in the hypothesis, which is not a feature of any individual link w_i but of the entire sentence hypothesis. λ_0 may be identified with the often used word insertion penalty. $C(\Lambda)$ is again a normalization factor that makes P_Λ a probability is the space of the hypotheses $\mathcal{H}$.

Note that in the simple case where the only features collected for the word-link w_i are the acoustic and language model log probabilities, (2) becomes the usual formulation for speech recognition which includes the language model scaling factor and the word insertion penalty.

1.3. Optimization of the Parameters Using Minimum WER Training

The model in (2) is used under the standard formulation for speech recognition which uses the Maximum a Posteriori Probability (MAP) rule to make a decision for the best hypothesis. Thus the empirical number of errors the model makes using a set of parameters Λ, is given by:

$$E(\Lambda) = \sum_{k=1}^{K} \mathcal{L}\left(\arg\max_{W \in \mathcal{W}_k} \log P_\Lambda(W|\mathcal{I}_k), W_{k_0}\right) \qquad (3)$$

where K is the number of utterances in our test data, W_{k_0} is the correct transcription for the kth utterance, $\mathcal{L}(\cdot,\cdot)$ is the Levenstein distance between two strings and $\mathcal{W}_k$ is the hypotheses space (typically a lattice or an N-best list) for the kth utterance.

The definition (2) is similar to the model used in the DMC approach [1]. In that case the features combined were log likelihoods from different acoustic and language models. The optimization techniques presented in [1] use smooth error measures to approximate the expected value of $E(\Lambda)$, which lead to gradient descent or closed form solution algorithms.

In this work we use the non-smooth empirical error count as the objective function to be directly minimized during optimization of the parameters Λ using the multidimensional simplex downhill method [4], also known as *amoeba search*. The method minimizes a function of n variables using a comparison of function values at the (n+1) vertices of a general simplex, followed by the replacement of the vertex with the highest value by another point. The simplex adapts itself to the local landscape, and contracts on to a local minimum. The method is effective and computationally compact, but it finds only a local minimum whose value depends on the original simplex. To overcome this effect, a variation of the algorithm is used, where the search keeps restarting with a mutation of the original simplex until the solution becomes stable.

2. EXPERIMENTS

2.1. Experimental Setup

The Switchboard (SWBD) database was used for the experiments described in this work. For the training of the parameters (the weights λ_j) we used ~3.5 hours of training data (4k sentences, 48k words). These were collected from utterances not used for training the acoustic model parameters. From the same set of utterances we set aside ~40min (724 sentences, 8k words) of test data (set test1). The model was also evaluated on the 2427 sentences that formed the dev-test at the 1997 Johns Hopkins University LVCSR Workshop (set test2). For the three data sets, 2000-best hypotheses were obtained from the baseline word-lattices and were used for training and rescoring with the new parameters.

2.2. Features Used

The features used in this experiments for the model (2) may be divided in four categories, depending on the source of information used to compute them:

1.BASE: Features used by the baseline system:

ac_score: log probability of the acoustic model, $\log P_{AC}(w_i)$.

lm_score: log probability of the language model, $\log P_{LM}(w_i|h_i)$.

num_words: the number of words in the whole hypothesis W.

2.VOTING: Features that are word-level votes (agreement counts) from the output of the recognizer (there is no use of model scores). We say that two links agree when they correspond to the same word in the vocabulary.

vote_start: log of the fraction of the paths that have links that agree with the current link w_i and start around the same time.

vote_end: log of the fraction of the paths that have links that agree with w_i and end around the same time.

vote_middle: log of the fraction of the paths that have links that agree with w_i at the middle time of the link.

	Train WER(%)	test1 WER(%)	test2 WER(%)
BASE	36.84	34.4	38.9
BASE+VOTING	36.72	34.3	38.8
BASE+LATTICE	36.60	34.2	38.6
BASE+LOCAL	36.35	33.6	38.3
FULL	35.93	33.4	38.1

Table 1: WER results

3.LATTICE: Features that are computed using information from the whole lattice structure about the reliability of the word associated with the link w_i. Comparisons between links aim to capture the source of this reliability (i.e whether it comes from the acoustic or the language model, or the combination of both).

forward_sc: log forward score at the beginning node of the link. This is the sum of the baseline system scores for all the partial paths in the lattice that end at that node. We normalize the score dividing by the time length from the beginning of the lattice to the node.

backward_sc: log backward score at the end node of the link. This is the sum of the baseline system scores for the partial paths that start at that node. We normalize by the time length from the node to the end of the lattice.

total_prob: log of the total posterior probability for the word associated with the link. This is the sum of the baseline system scores of all the paths that go through links that overlap with w_i and have the same word, normalized by the total combined score of the lattice.

total_ac: as above but only P_{AC} scores are used to find the total acoustic probability.

total_lm: as above but only P_{LM} scores are used.

tot_prob_ratio: difference between *total_prob* score for link w_i and the parallel link with the highest *total_prob*.

tot_ac_ratio: difference between *total_ac* score for w_i and the parallel link with the highest *total_ac*.

tot_lm_ratio: difference between the *total_lm* score for w_i and the parallel link with the highest *total_lm*.

4.LOCAL: Features obtained using information associated with the time segment (s_i, e_i) and the word h_i which aim to capture local information about the acoustic reliability of the word. Some of these features use the output of a phone recognizer, which was implemented with the use of a phone trigram and the baseline acoustic triphone models.

ac_per_frame: average per frame acoustic score.

duration: expected acoustic duration of the word, computed using the transition probabilities of the acoustic models and the baseform pronunciation.

num_prons: log number of alternative pronunciations.

num_phones: number of phones in the word baseform pronunciation p.

n_diff_phrec: difference in number of phones between p and the phone recognition output.

ac_diff_phrec: difference of *ac_score* from the acoustic score of the phone recognition output.

ph_dist_phrec: the Levenstein phone distance between p and the phone recognition output.

ph_dist_achigh: the Levenstein phone distance between p and the pronunciation of the link with the highest *ac_per_frame* that occurs parallel to link w_i.

2.3. Results

We carried out five optimization experiments, and the results are presented in Table 1. The first four experiments correspond to each of the feature classes described in the previous section, and the last one is the FULL system which combines all of the above features in one model.

For the BASE experiment the same features were used as in the baseline system, with the parameters (weights) optimized using the training data. For the other three experiments the BASE features were combined with the features of one of the described classes.

The parameters were trained to minimize the empirical error $E(\Lambda)$ induced by the model, computed from the 2000-best lists[1] of the training data. The optimization

[1] The 2000-best lists are obtained using the baseline system.

algorithm used amoeba search, with the baseline solution as a point in the starting simplex. The results in Table 1 are obtained from rescoring the training and test 2000-best lists[1]with the new parameters.

We find that each of the feature sets is able to offer a small amount of improvement to the system. The LOCAL feature set gives the bigger gain by itself. Combining all the features in the FULL system (21 parameters) we achieve a significant WER improvement

3. CONCLUSIONS

We have introduced a new method of using word-level features with the goal of improving speech recognition accuracy. These features are used to define a new likelihood score for each hypothesized utterance. A small number of parameters is optimized to minimize the number of errors induced when the model is used by a MAP decoder.

By adding 21 new parameters to our system we were able to achieve 1.0% WER reduction on the test1 data and 0.8% reduction on test2.

4. ACKNOWLEDGMENTS

This work was influenced by useful discussions with Peter Beyerlein. The author also thanks Fred Jelinek for support and comments on this work, and the colleagues in CLSP for help with the preparation of this paper.

References

1. P. Beyerlein. Discriminative model combination. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.
2. Lin Chase. *Error-Responsive Feedback Mechanisms for Speech Recognizers.* PhD thesis, CMU, 1997.
3. Jonathan G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *ASRU*, 1997.
4. J. A. Nelder and R. Mead. A simplex method for function minimization. In *Computer Journal*, volume 7, pages 308–313, 1965.
5. Thomas Schaaf and Thomas Kemp. Confidence measures for spontaneous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1997.
6. D. Vergyri. Use of word level side information to improve speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000.